Introduction to LAN/WAN

Network Layer (part II)
Topics

- The Network Layer
  - Introduction
  - Routing (5.2)
  - The Internet (5.5)
    - IP, IP addresses
    - ARP (5.5.4)
    - OSPF (5.5.5)
    - BGP (5.5.6)
  - Congestion Control (5.3)
Internetworking

- Internet: different small networks connected
- Different Protocols (TCP/IP, SNA, Appletalk, NCP/IPX, etc)
Internet Structure

- **Backbones**: high bandwidth lines, fast routers
- **Regional networks** attached to backbones
- **LANs** (universities, companies, ISP, etc) connected to regional network
Internet Protocol (IP)

- IP concerned with routing (best effort)
- Interesting options:
  - security, strict source routing, loose source routing, record route, timestamp

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of service</th>
<th>Total length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Identification</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D F M</td>
<td>Fragment offset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time to live</td>
<td>Protocol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Header checksum</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Source address</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Destination address</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options (0 or more words)</td>
<td></td>
</tr>
</tbody>
</table>
Internet Protocol (IP)

Summary of options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Specifies how secret the datagram is</td>
</tr>
<tr>
<td>Strict source routing</td>
<td>Gives the complete path to be followed</td>
</tr>
<tr>
<td>Loose source routing</td>
<td>Gives a list of routers not to be missed</td>
</tr>
<tr>
<td>Record route</td>
<td>Makes each router append its IP address</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Makes each router append its address and timestamp</td>
</tr>
</tbody>
</table>
IP Addresses

IPv4: 32 bit addresses: $2^{32}$ addresses (e.g. ccc4.wpi.edu has IP address 130.215.36.158)

Addresses controlled by ICANN

Previously classfull addressing (A, B, C, etc)
- Class A: fix first bit at 0, $2^7$ networks, $2^{24}$ hosts
- Class B: start with 10, $2^{14}$ networks, $2^{16}$ hosts, etc
- Class C: start with 110, $2^{29}$ networks, $2^8$ hosts
IP Addresses

- Disproportional demands for address classes:
  - Few organizations have up to $2^{24}$ hosts (class A)
  - Most want class B ($2^{16}$ hosts)
  - Quick depletion of some classes (class B) while others remained un-used (class A)

- Now Classless InterDomain Routing (CIDR) pronounced “cydar”
  - Basically allow variable network address (subnet mask) and host address part
  - Indicate length by adding subnet mask
  - Example: 223.1.1.0/24 means first 24 bits should be treated as network address (subnet mask)

- CIDR was temporary solution => IPv6 (128 bits)
IP Addresses

Another problem:
- ISP may have one class B address (/16) address
- This equals 65,534 host numbers
- Dial-up customers: assign IP addresses temporarily
- Permanent: can assign maybe 1 or 2 per customer
- However, small businesses may have many machines
- Also, now many homes have multiple machines with DSL or cable (always on!!)
- Solution: Network Address Translation
- Basic idea:
  - each machine within small business or home gets new unique IP address within its network
  - NAT box translates all addresses to one actual address when going out
Network to Data Link Address Translation

- Internet hosts use IP
- Data link layer does not understand IP
  - Ethernet uses 48-bit address
  - ex: ifconfig gives 00:10:4B:9E:B3:E6

Q: How do IP addresses get mapped onto data link layer addresses, such as Ethernet?
A: The Address Resolution Protocol (ARP)
Example 1

Host 1 wants to send a message to Host 2, say “mary@eagle.cs.uni.edu”
Address Resolution

- Lookup IP of *eagle.cs.uni.edu*
  - DNS (chapter 7)
  - returns 192.31.65.5

- Host 1 builds packet to 192.31.65.5
  - now, how does data link layer know where to send it?
  - need Ethernet address of Host 2

- Could have config file to map IP to Ethernet
  - hard to maintain for thousands of machines
Address Resolutioning

- Host 1 broadcasts packet on LAN with IP address 192.31.65.0 asking “Who owns IP address 192.31.65.5?”
- Each machine checks its IP address.
- Host 2 responds w/ Ethernet address (E2) – *Address Resolution Protocol* (ARP)
- Host 1 data-link can then encapsulate IP packet in frame addressed to E2 and dump
- Ethernet board on Host 2 recognizes, strips frame header and sends up to IP layer
ARP Optimizations

Send to H2 again?
- cache requests (time out in case of new card)

Many times, H1 requires ack from H2
- send H1 IP + Ethernet (192.31.65.7, E1)
- H2 caches and uses if needed

Hosts broadcast mapping when boot
- host looks for its own IP address
  - should get no answer, else don’t boot
- other Ethernet hosts all cache answer
Example 2

Host 1 sends message to Host 4

Problem: Router does not forward data-link layer broadcasts
Solutions

_solution_ 1

- CS router configured to respond to ARP requests for 192.31.63.0
- Host 1 makes an ARP cache entry of (192.31.63.8, E3)
  - sends all traffic to Host 4 to CS router
- Called Proxy ARP

_solution_ 2

- Host 1 knows Host 4 is on different subnet
  - sends to CS router
- CS router doesn’t need to know about remote networks
Either way ...

- Host 1 packs IP into Enet frame to E3
- CS router receives frame, removes packet
  - sees 192.31.63.0 to 192.31.60.7
- Sends ARP packet onto FDDI
  - learns 192.31.60.7 is at F3
- Puts packet into payload of FDDI frame and put on ring
- EE router receives frame, removes packet ...
Inside Out and Upside Down

Can a host learn its IP address at boot?

Unreasonable? No!! diskless workstation

- *Reverse Address Resolution Protocol (RARP)*

Broadcast:

- "my enet adress 13.05.05.18.01.25"
- "does anyone know my IP?"

RARP server sees request, sends IP

RARP broadcasts not across router

- BOOTP uses UDP

BOOTP requires sys admin to manually enter (IP address, Ethernet Address) in server

- Dynamic Host Configuration Protocol (DHCP) allows automatic, timeouts, recovery if host leaves, etc
Routing on the Internet

- Internet made up of Autonomous Systems (AS)
- Standard for routing inside AS
  - Interior Gateway Protocol
  - OSPF
- Standard for routing outside AS
  - Exterior Gateway Protocol
  - BGP
ASes, Backbones and Areas
Open Shortest Path First (OSPF)

- 1979, RIP (distance vector), replaced by link-state (Dijkstra)
- In 1990, OSPF standardized
- “O” is for “Open”, not proprietary
- ASes can be large, need to scale
  - Areas, that are self-contained (not visible from outside)
OSPF, continued

- Every AS has a *backbone*, area 0
  - all areas connect to backbone, possibly by a tunnel
- Routers are nodes and links are arcs with weights
- Computes “shortest” path for each:
  - delay
  - throughput
  - reliability
- Floods link-state packets
Border Gateway Protocol (BGP)

- Inside AS, only efficiency
- Between AS, have to worry about politics
  - No transit traffic through some ASes
  - Never put Iraq on a route starting at the Pentagon
  - Do not use the US to get from British Columbia to Ontario
  - Traffic starting or ending at IBM should not transit Microsoft
- BGP router pairs communicate via TCP
  - hides details in between
- Uses distance vector protocol
  - but “cost” can be any metric
When too much traffic is offered, congestion sets in and performance degrades sharply.
Causes of Congestion

- Queue build up until full
  - Many input lines to one output line
  - Slow processors
  - Low-bandwidth lines
    - system components mismatch (bottleneck)
  - Insufficient memory to buffer
- If condition continues, infinite memory makes worse!
  - timeouts cause even more transmission
  - congestion feeds upon itself until collapse
Flow Control vs. Congestion Control

- Congestion control (network layer)
  - make sure subnet can carry offered traffic
  - global issues, including hosts and routers
- Flow control (data link layer)
  - point-to-point between sender and receiver
  - fast sender does not overpower receiver
  - involves direct feedback to sender by receiver
- Some congestion solutions:
  - Choke packets
  - Traffic Shaping (leaky bucket)